CLAIMS

What is claimed is:

1. A method for maintaining the reservation state in a network router, comprising maintaining a bounded aggregate per-destination reservation state instead of a per-flow reservation state.

2. A method as recited in claim 1:

wherein said router maintains rates of incoming and outgoing traffic; and

wherein said router does not maintain information on rates of flow.

3. A method as recited in claim 1, further comprising maintaining a set of token-buckets arranged in the form of a tree for aggregating network flows into classes.

4. A method as recited in claim 1, further comprising aggregating network flows utilizing burst-drain-time or burst-ratio.

5. A method as recited in claim 1, further comprising merging a set of data flows into a smaller set of aggregated flows.

6. A method as recited in claim 5, wherein said data flows are merged based on class or destination.

7. A method as recited in claim 1, wherein said router maintain state only for aggregated flows and processes only aggregated flows.

8. A method as recited in claim 7, further comprising:

providing guarantees to aggregated flows; and

providing guarantees to individual flows within the aggregated flows.

9. A method as recited in claim 1, further comprising using diffusing computations to maintain consistency of the reservation.

10. A method as recited in claim 1, wherein said aggregate state has a size and associated refresh mechanism.

11. A method as recited in claim 10, wherein aggregate state size and refresh mechanism complexity are a function of a network parameter rather than a function of the number of end-user flows.

12. A method as recited in claim 11, wherein said network parameter comprises class.

13. A method as recited in claim 11, wherein said network parameter comprises destination.

14. A method as recited in claim 1, wherein said step of maintaining aggregate per-destination reservation state instead of a per-flow reservation state comprises storing and refreshing resource reservations on a per-destination basis rather than on a per-flow basis.

15. A method as recited in claim 1, further comprising utilizing per-destination refresh messages instead of per-flow refresh messages.

16. A method as recited in claim 15, wherein a refresh message specifies a destination and bandwidth for that destination.

17. A method as recited in claim 16, wherein when a flow terminates, the source stops sending refresh messages and bandwidth reserved for the flow is released.

18. A method as recited in claim 1:

wherein the source of a flow sends a refresh message to said router; and

wherein all refresh messages of a particular destination are aggregated at said router.

19. A method as recited in claim 18, wherein a refresh message specifies a destination and bandwidth for that destination.

20. A method as recited in claim 19, wherein when a flow terminates, the source stops sending refresh messages and bandwidth reserved for the flow is released.

21. A method for maintaining the reservation state in a network router, comprising storing and refreshing resource reservations on a per-destination basis, rather than on a per-flow basis.

22. A method as recited in claim 21:

wherein said router maintains rates of incoming and outgoing traffic; and

wherein said router does not maintain information on rates of flow.

23. A method as recited in claim 21, further comprising maintaining a set of token-buckets arranged in the form of a tree for aggregating network flows into classes.

24. A method as recited in claim 21, further comprising aggregating network flows utilizing burst-drain-time or burst-ratio.

25. A method as recited in claim 21, further comprising merging a set of data flows into a smaller set of aggregated flows.

26. A method as recited in claim 25, wherein said data flows are merged based on class or destination.

27. A method as recited in claim 21, wherein said router maintain state only for aggregated flows and processes only aggregated flows.

28. A method as recited in claim 27, further comprising:

providing guarantees to aggregated flows; and

providing guarantees to individual flows within the aggregated flows.

29. A method as recited in claim 21, further comprising using diffusing computations to maintain consistency of the reservation.

30. A method as recited in claim 21, wherein said aggregate state has a size and associated refresh mechanism.

31. A method as recited in claim 30, wherein aggregate state size and refresh mechanism complexity are a function of a network parameter rather than a function of the number of end-user flows.

32. A method as recited in claim 31, wherein said network parameter comprises class.

33. A method as recited in claim 31, wherein said network parameter comprises destination.

34. A method as recited in claim 21, further comprising utilizing per-destination refresh messages instead of per-flow refresh messages.

35. A method as recited in claim 34, wherein a refresh message specifies a destination and bandwidth for that destination.

36. A method as recited in claim 35, wherein when a flow terminates, the source stops sending refresh messages and bandwidth reserved for the flow is released.

37. A method as recited in claim 21:

wherein the source of a flow sends a refresh message to said router; and

wherein all refresh messages of a particular destination are aggregated at said router.

38. A method as recited in claim 37, wherein a refresh message specifies a destination and bandwidth for that destination.

39. A method as recited in claim 38, wherein when a flow terminates, the source stops sending refresh messages and bandwidth reserved for the flow is released.

40. A method for maintaining the reservation state in a network router, comprising:

merging a set of data flows into a smaller set of aggregated flows; and

storing and refreshing resource reservations on a per-destination basis, rather than on a per-flow basis;

wherein said router maintains rates of incoming and outgoing traffic; and

wherein said router does not maintain information on rates of flow.

41. A method as recited in claim 40, further comprising maintaining a set of token-buckets arranged in the form of a tree for aggregating network flows into classes.

42. A method as recited in claim 40, further comprising aggregating network flows utilizing burst-drain-time or burst-ratio.

43. A method as recited in claim 40, wherein said data flows are merged based on class or destination.

44. A method as recited in claim 40, wherein said router maintains state only for aggregated flows and processes only aggregated flows.

45. A method as recited in claim 44, further comprising:

providing guarantees to aggregated flows; and

providing guarantees to individual flows within the aggregated flows.

46. A method as recited in claim 40, further comprising using diffusing computations to maintain consistency of the reservation.

47. A method as recited in claim 40, wherein said aggregate state has a size and associated refresh mechanism.

48. A method as recited in claim 47, wherein aggregate state size and refresh mechanism complexity are a function of a network parameter rather than a function of the number of end-user flows.

49. A method as recited in claim 48, wherein said network parameter comprises class.

50. A method as recited in claim 48, wherein said network parameter comprises destination.

51. A method as recited in claim 40, further comprising utilizing per-destination refresh messages instead of per-flow refresh messages.

52. A method as recited in claim 51, wherein a refresh message specifies a destination and bandwidth for that destination.

53. A method as recited in claim 52, wherein when a flow terminates, the source stops sending refresh messages and bandwidth reserved for the flow is released.

54. A method as recited in claim 40:

wherein the source of a flow sends a refresh message to said router; and

wherein all refresh messages of a particular destination are aggregated at said router.

55. A method as recited in claim 54, wherein a refresh message specifies a destination and bandwidth for that destination.

56. A method as recited in claim 55, wherein when a flow terminates, the source stops sending refresh messages and bandwidth reserved for the flow is released.